VŠB-Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Grafická uživatelská aplikace pro tvorbu bezkontaktních schémat

The Graphical User Application for Logic Circuits Design

Student **Ngo Manh Luong**

Vedoucí bakalářské práce **Ing. Jolana Škutová**

Otrava 2009

Prohlášení studenta

Prohlašuji, že jsem celou diplomovou (bakalářkou) práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové (bakalářské) práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne: .....

.....

Popis studenta

Prohlašuji, že

- Byl jsem seznámen s tím, že na moji diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. - autorský zákon -,zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo
- Beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB – TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou (bakalářskou) práci užít (§ 35 odst. 3)
- Souhlasím s tím, že jeden výtisk diplomové (bakalářské) práce bude uložen v ústřední knihově VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO
- Bylo sjednáno, že s VŠB-TUO,v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona
- Bylo sjednáno, že užít své dílo – diplomovou (bakalářskou) práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO,která je oprávněna v takovém případě ode mne požadovat přimčený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- Beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č.111/1998 Sb. o vysokých školách a o změně a doplněn dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě dne:..... Popis studenta: .....

ANOTACE DIPLOMOVÉ PRÁCE

Luong, N. M. Grafická uživatelská aplikace pro tvorbu bezkontaktních schémat. Ostrava: Katedra automatizační techniky a řízení, Fakulta strojní, VŠB - Technická Univerzita Ostrava, 2009, 41 s. Bakalářská práce, vedoucí Škutová, J.

Bakalářská práce se zabývá realizací grafické uživatelské aplikace v prostředí Matlab, která umožňuje generování bezkontaktních schémat pomocí NAND nebo NOR logických prvků.

Teoretický úvod práce stručně popisuje problematiku kombinační logiky a na vybraných příkladech uvádí postup tvorby bezkontaktních schémat s využitím pouze NAND nebo pouze NOR prvků se dvěma vstupy (v program Simulink). Další část práce je zaměřena na popis GUI aplikace v programu Matlab a v návaznosti pak konkrétní návrh grafické uživatelské aplikace a základní časti programového kódu pro události řízené prvky aplikace.

ANOTATION OF THESIS

Luong, N. M. Graphical user application for creating contactless schemes. Ostrava: Applied Informatics and Control, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 2009. 41 p. Bachelor thesis, Head: Škutová, J.

Thesis deals with the realize of graphical user applications in Matlab, which allows the generation of contactless schemes using NAND or NOR logic elements.

Theoretical introduction briefly describes the work of combinational logic problems and selected examples of states making process contactless schemes using only NAND or NOR elements with two inputs (in the Simulink). Another part of the work is focused on the description of GUI applications in Matlab, and following the concrete draft applications and graphical user base of the program code for event driven elements of applications.

Obsah

Seznam použitých zkratek a symbolů

AND	Logická funkce vyjadřující logický součin
GUI	Grafické uživatelské rozhraní
KLO	Kombinační logický obvod
NOT	Logická funkce vyjadřující logický negaci
NAND	Logická funkce vyjadřující negovaný logický součin
NOR	Logická funkce vyjadřující negovaný logický součet
OR	Logická funkce vyjadřující logický součet
XOR	Logická funkce vyjadřující exkluzivní (výhradní) součet

1. Úvod

Automatické řízení je předmět potřební k logické dedukci. Studenti se stým to obtížným předmětem Základy automatizace seznámí už v prvním ročníku. Student v tomto předmětu realizuje program pro generování schémat realizací logického řízení pomocí NAND nebo NOR logických prvků v prostředí Matlab, vizuální ověření minimalizace logické funkce zadáváním libovolných kombinací vstupů. Důraz na návaznost aplikovatelnosti pro reálný systém řízení. Instalace MATLAB je velmi jednoduchá a napomáhá s propojením s vyšším jazykem (kódy). A jinak studenti budou používat často v předmětu automatické řízení, a pak s tímto programem můžu uvést oba dva: Matlab i způsoby realizací schémat logického řízení.

V první části jsem vysvětlil kombinační logiku, která obsahuje logické funkce, logické zákony, minimalizace logické funkce, tvorbu schémat a kombinační logické obvody. Dále jsem do své práce zařadil schémata NAND a NOR pro vybrané příklady. Vztahy schémat NAND a NOR v okně Simulink a výsledné tabulky.

V druhé části jsem se zabýval tvorbou aplikací v programu Matlab/Guide. Popisuji grafické uživatelské prostředí MATLAB/GUIDE a funkci v souvislosti s problematikou generovaní bloků Simulink.

V třetí části je navržena a realizována uživatelská aplikace pro tvorbu schémat NAND a NOR. Představil jsem uživatelské funkční aplikace a vysvětlil, jak se používají. Pak jsem uvedl, ovládací prvky i kódů pro uživatelské funkce. Také jsem uvedl a vysvětlil každý povinný kód.

Ve čtvrté časti je provedeno hodnocení dosažených výsledku a navržen další směr k řešení problematiky

2. Logické kombinační obvody

Obor logika, výroková logika nebo matematická logika souvisí s oborem dvouhodnotová proměna. Na začátku oboru logiky byla Boolova algebra, kterou sestavil v roce 1830 George Boole.Boolova, algebra je algebraický systém, jehož oborem hodnot jsou pouze dvě hodnoty, ve většině případů se uvažují hodnoty 0 a 1 (někdy se mohou označovat taky jako hodnoty True, False), a množina operací nad tímto oborem hodnot, které se také označují jako logické funkce [Wikipedia].

2.1 Logické funkce

Logické funkce využívají několika vstupních proměnných označeních nejčastěji abecedně, tedy vstupy a, b, c, d .... a výstupní proměnné označované y. Logické funkce NOT, AND, OR, NAND, NOR, XOR jsou základními všeobecnými známými logickými funkcemi.

Logické funkce mohou být zapsány logickým výrazem (logické proměnné spojené operátory základních logických funkcí, pravdivostní tabulkou nebo Karnaughovou mapou (zpravidla se využívá pro minimalizace logických funkcí).

Tab. 1 Pravdivostní tabulka logické funkce NOT

a	*y*
0	*1*
1	*0*

V případě logické funkce NOT se uvažuje jeden vstup a logická funkce je definována logickým výrazem

$$y = \bar{a} \tag{1}$$

Pro logickou funkci NOT je pravdivostní tabulka uvedena v logické funkce využívají několika vstupních poměrných označeních nejčastěji abecedně tedy vstupy a, b, c, d.... a výstupní promění označované y logické funkce not, and, or, nand, nor, or ,xor jsou základními všeobecnými známými logickými funkcemi. Logické funkce mohou být zapsány logickým výrazem (logické proměnné spojené operátory základních logických funkcí, pravdivostní tabulkou nebo karnaughovou mapou (zpravidla se využívá pro minimalizace logických funkcí).

Z důvodu zaměření bakalářské práce na realizace jakýchkoliv logických funkcí pomocí bezkontaktních schémat, které zahrnují pouze logické prvky NAND nebo NOR se dvěma vstupy, jsou dále uvedeny popisy logických funkcí NAND a NOR.

Logická funkce NAND (negovaný logický součin – Scheferova funkce) je dána vztahem

$$y = \overline{a \cdot b} \tag{2}$$

a pravdivostní tabulka pro logickou funkci NAND je uvedena v tab. 2

Tab. 2 Pravdivostní tabulka logické funkce NAND

a	*b*	*y*
0	*0*	*1*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

Logická funkce NOR (negovaný logický součet – Piercerova funkce) je dána vztahem

$$y = \overline{a + b} \tag{3}$$

A pravdivostní tabulka pro logickou funkci NOR je uvedena v tab. 3

Tab. 3 Pravdivostní tabulka funkce NOR

a	*b*	*y*
0	*0*	*1*
0	*1*	*0*
1	*0*	*0*
1	*1*	*0*

Obr. 1. Značení logických prvků NAND (vlevo) a NOR (vpravo)

V oblasti realizace logických funkcí se vyžívají grafické symboly odpovídající logickým funkcím. Pro funkci NAND a NOR jsou grafické symboly na obr. 1

2.2 Logické zákony

Logický zákon	Zápis logického zákona
1. Zákon vyloučení třetího	$a * \bar{a} = 0$ $a + \bar{a} = 1$
2. Zákon agresivity	$a * 0 = 0$ $1 + a = 1$
3. Zákon dvojité negace	$\bar{\bar{a}} = a$
4. Zákon absorpce	$a + a = a$ $a * a = a$
5. Komutativní zákon	$a + b = b + a$ $a * b = b * a$
6. Asociativní zákon	$a + (b + c) = (a + b) + c$ $a * (b * c) = (a * b) * c$
7. Distributivní zákon	$a * (b + c) = a * b + b * c$ $a + b * c = (a + b) * (a + c)$
8. Absorpční zákon (zákon agrese)	$a + \bar{a} * b = a + b$ $a * (\bar{a} + b) = a * b$
9. Neutrálnost	$0 + a = a$ $1 * a = a$
10. De morganovy zákony	$\overline{a + b} = \bar{a} \cdot \bar{b}$ $\overline{a \cdot b} = \bar{a} + \bar{b}$

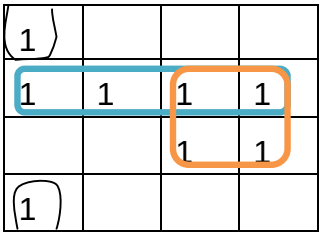
Tab. 4 Souhrn logických zákonů a jejich vyjádření logickou rovnici

Pro operace s logickými proměnnými a výraz platí obdobně jako při aritmetických operacích určité zákony. Hovoříme o zákonech formální logiky, nebo také o zákonech Booleovy algebry (viz tab. 4)

2.3 Minimalizace logické funkce a tvorba schémat

K dané logické funkci existuje několik různých tvarů. Všechny jsou matematicky stejné, protože představují stejnou funkční závislost, i když mohou být tvarově odlišně značeny. Nejsou však rovnocenné z hlediska technického. Pro technickou realizaci je nutno vždy funkci upravit do nejjednoduššího tvaru – minimalizovat ji. Minimalizaci funkce lze dosáhnout toho, že při její realizaci je nutný nejmenší počet logických prvků (negací, konjunkcí, disjunkcí). Tím se logický obvod stane jednoduchým.

Minimalizace je v podstatě postup, jimž se hledá minimální normální forma dané funkce, toho lze dosáhnout podle níže popsané metody, toho lze dosáhnout podle níže popsané metody.



Obr. 2.Karnauhova mapa pro logickou funkci (viz 2.5.1)

Karnaughova mapa je tabulka (obr. 2), která má tolik políček, kolik je kombinací vstupních proměnných – (kde n je počet vstupních proměnných). Každé políčko odpovídá jedné z možných kombinací a zapisujeme do něj odpovídající funkční hodnotu. Podle kódu, kterým přiřazujeme políčka jednotlivým kombinacím proměnných, rozlišujeme různé mapy. Nejpoužívanější je Karnaughova mapa. Zde se sousední políčka od sebe liší hodnotu jediného argumentu. Do mapy zapisuje pouze 1 nebo 0.

V Karnaughova mapě můžeme sdružovat 2, 4, 8 ... sousední políčka, která mají hodnotu 1 (obecně , kde n=0, 1, 2, 3 ... – to znamená, že samostatnou jedničku musíme také použít, protože).

Pravidlem pro sdružování políček je smyčkové pravidlo tvorby co nejmenšího počtu smyček a zároveň co největšího seskupení jedniček ve smyčce. Smyčka může sdružovat pouze jeničky (pro úplnou disjunktivní normální formu – ÚNDF, tato se předpokládá pro využití v celé bakalářské práci).

2.4 Kombinační logické obvody

Logický obvod je fyzikální systém, který realizuje logické operace. Je realizován skupinou logických členu vzájemně spojených tak, aby realizovaly žádané logické funkce. Logický obvod je takový obvod, u něhož může každá veličina ve vstupu i výstupu v ustáleném stavu nabývat s určenou přesností jen jednu ze dvou možných hodnot.[Logické řízení]

Kombinační logické obvody (KLO) představují takové logické obvody, jejichž stavy ve výstupech závisí pouze na okamžitých kombinacích vstupních proměnných a odpovídá jediná hodnota výstupního logického obvodu. Kombinační logické obvody nemají žádnou paměť předchozích stavů, tuto schopnost mají sekvenční logické obvody a tato problematika není náplní bakalářské práce.

Pro realizaci kombinačních logických obvodů je možné použít pevné paměti, programovatelná logická pole a kombinačními obvody NAND a NOR lze realizovat základní logické funkce NOT, AND nebo OR a také libovolné funkce po nezbytných úpravách s využitím logických De Morganových zákonů.[Chlebek,2007]

3. Realizace schémat NAND a NOR v prostředí Matlab/Simulink

Pro seznámení s tvorbou schémat a sestavování bloků, nastavení jejich parametrů, které předchází tvorbě uživatelské aplikací v prostředí GUI, byly zadány konkrétní příklady pro realizaci schémat pomocí funkce NAND nebo schéma pomocí funkce NOR. Zadané příklady jsou logické funkce, které mají čtyři vstupní proměnné a, b, c, d, jednu výstupní proměnou y. Než sestavíme bezkontaktní schéma s prvky NAND nebo schéma s prvky NOR, je nutné upravit původní tvar logické funkce a v následujících podkapitolách je uveden postup úpravy zadaných logických funkcí.

3.1 Logická funkce – Příklad 1

Logická funkce je dána vztahem

$$y = b(a + \bar{c}) + \bar{b}c\bar{d} \qquad (4)$$

Funkci vyjádřenou základními logickými funkcemi je nutné převést pomocí De Morganových logických zákonů

$$\overline{a \cdot b} = \bar{a} + \bar{b} \qquad (5)$$

Na funkci vhodnou pro realizaci pouze prvky NAND se dvěma vstupy a tvar upravené funkce je dán vztahem

$$y = \overline{\overline{b \cdot \bar{a}\bar{c}} \cdot \overline{\bar{b}c\bar{d}}} \qquad (6)$$

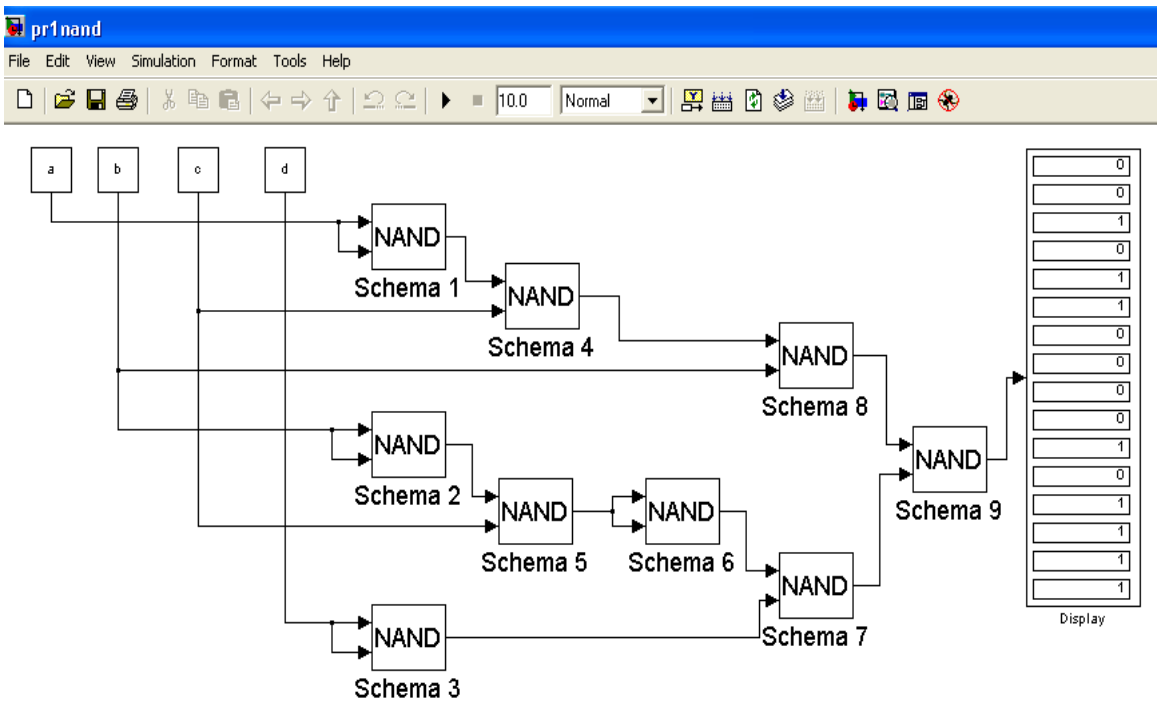
Další úprava pro realizaci schémat s využitím NOR prvků je obdobná, vychází z De Morganova logického zákona

$$\overline{a + b} = \bar{a} \cdot \bar{b} \qquad (7)$$

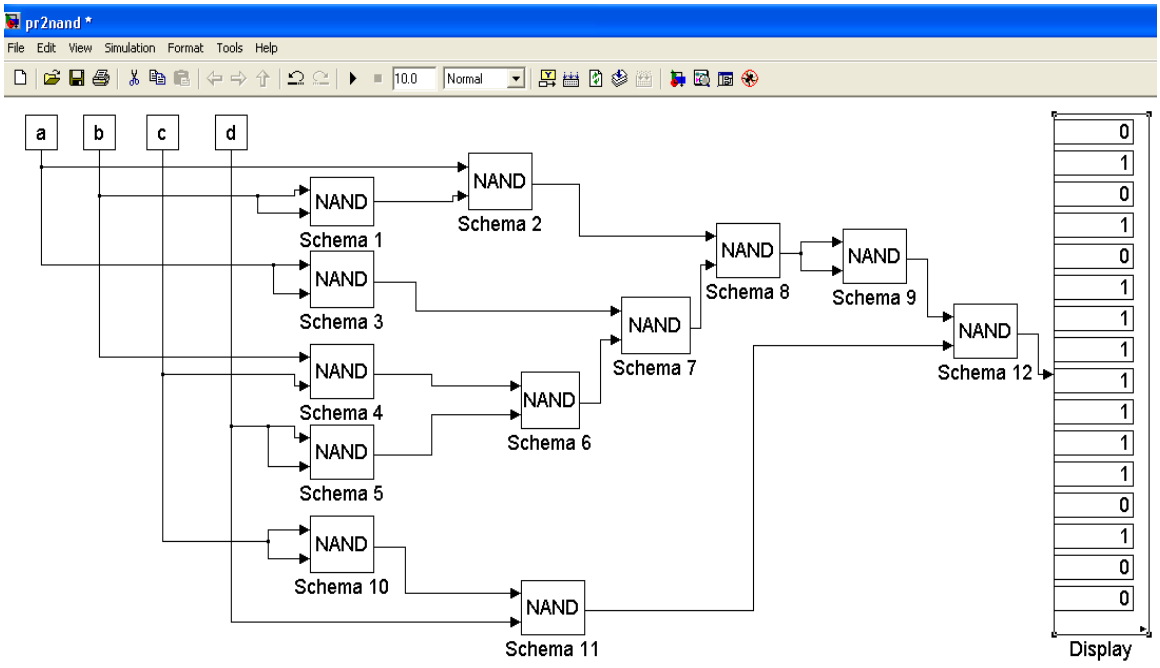
a upravená logická funkce vhodná pro realizaci pouze prvky NOR je dána vztahem

$$y = \overline{\bar{b} + \overline{a + \bar{c}}} + \overline{b + \bar{c} + d}\,. \qquad (8)$$

Pro logickou funkci byly vytvořeny z konkrétních bloků schémata realizací z NAND prvků (obr. 3) a schéma z NOR prvků (obr. 4).



Obr. 3. Příklad 1 – schéma z NAND prvků



Obr. 4. Příklad 1 – schéma z NOR prvků

Pravdivostní tabulka pro logickou funkci (4) musí odpovídat hodnotám zobrazeným v bloku Display, viz (obr. 3) a (obr. 4).

Tab. 5 Pravdivostní tabulka pro logickou funkci viz (4)

a	b	c	d	y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

3.2 Logická funkce – Příklad 2

Obdobně jako v kapitole 3.3.1. i zde byla zvolena logická funkce, pro kterou má být provedena úprava dle logických zákonů a tato funkce má být realizována prostřednictvím bez kontaktních schémat NAND a NOR. Logická funkce je dána vztahem

$$y = a\bar{b} + \bar{a}(bc + d) + \bar{c}d \qquad (9)$$

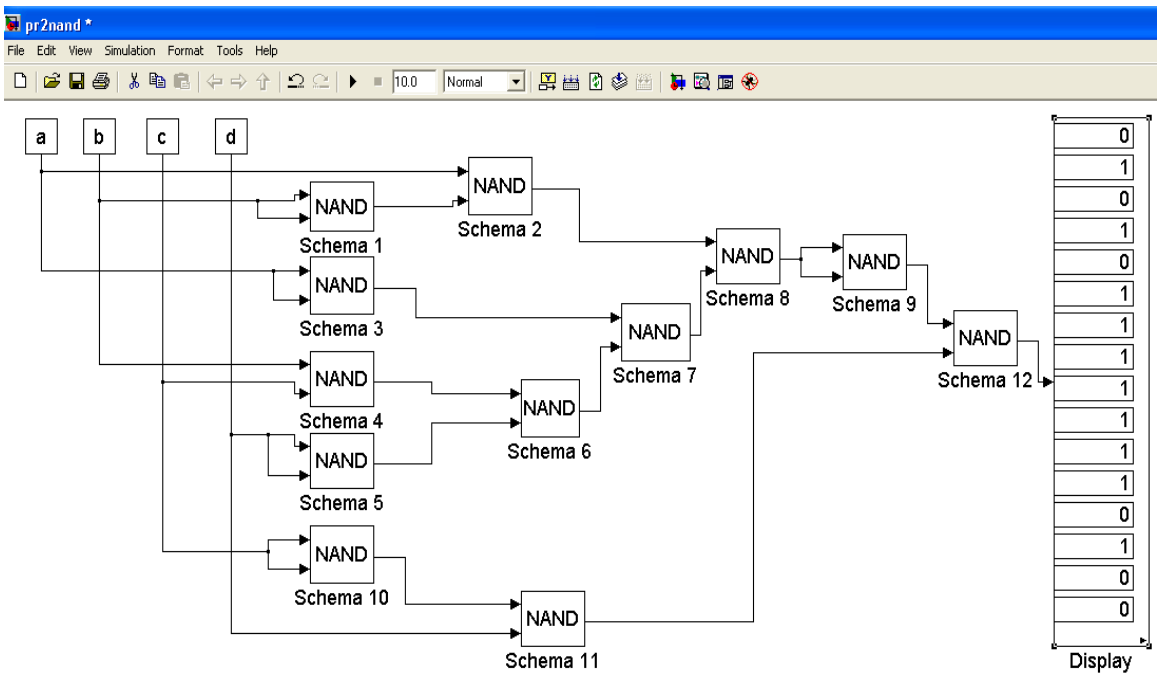
A její úprava pro realizaci bezkontaktního schéma s prvky NAND je dána takto

$$y = \overline{\overline{a\bar{b}} \cdot \overline{\bar{a}bc} \cdot \overline{\overline{\bar{d}} \cdot \overline{\bar{c}d}}} \qquad (10)$$

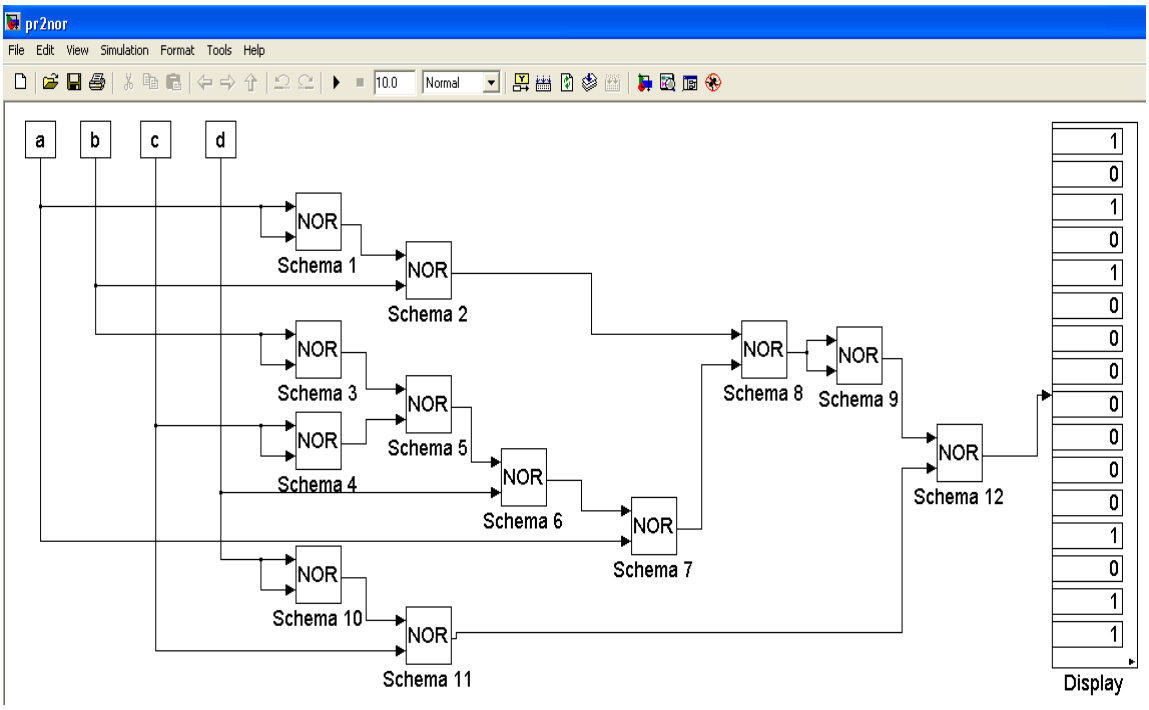
A úprava pro realizaci bezkontaktního schéma s prvky NOR je obdobně upravena jako

$$y = \overline{\bar{a} + b} + \overline{a + \overline{\overline{b} + \bar{c} + d} + \overline{c + \bar{d}}} \qquad (11)$$

Pro logickou funkci byly vytvořeny z konkrétních bloků schémata realizací z NAND prvků (obr. 5) a schéma z NOR prvků (obr. 6).



Obr. 5. Příklad 1 – schéma z NOR prvků



Obr. 6. Příklad 1 – schéma z NOR prvků

Pravdivostní tabulka pro logickou funkci (9) musí odpovídat hodnotám zobrazeným v bloku Display, viz (obr. 5) a (obr. 6).

Tab. 6 Pravdivostní tabulka pro logickou funkci viz (9)

a	b	c	d	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

3.3 Logická funkce – Příklad 3

Obdobně jako v kapitole 3.3.1. a 3.3.2 i zde byla zvolena logická funkce, pro kterou má být provedena úprava dle logických zákonů a tato funkce má být realizována prostřednictvím bezkontaktních schémat NAND a NOR. Logická funkce je dána vztahem

$$y = ad + \bar{a}(\bar{b}c + d) + bc \qquad (12)$$

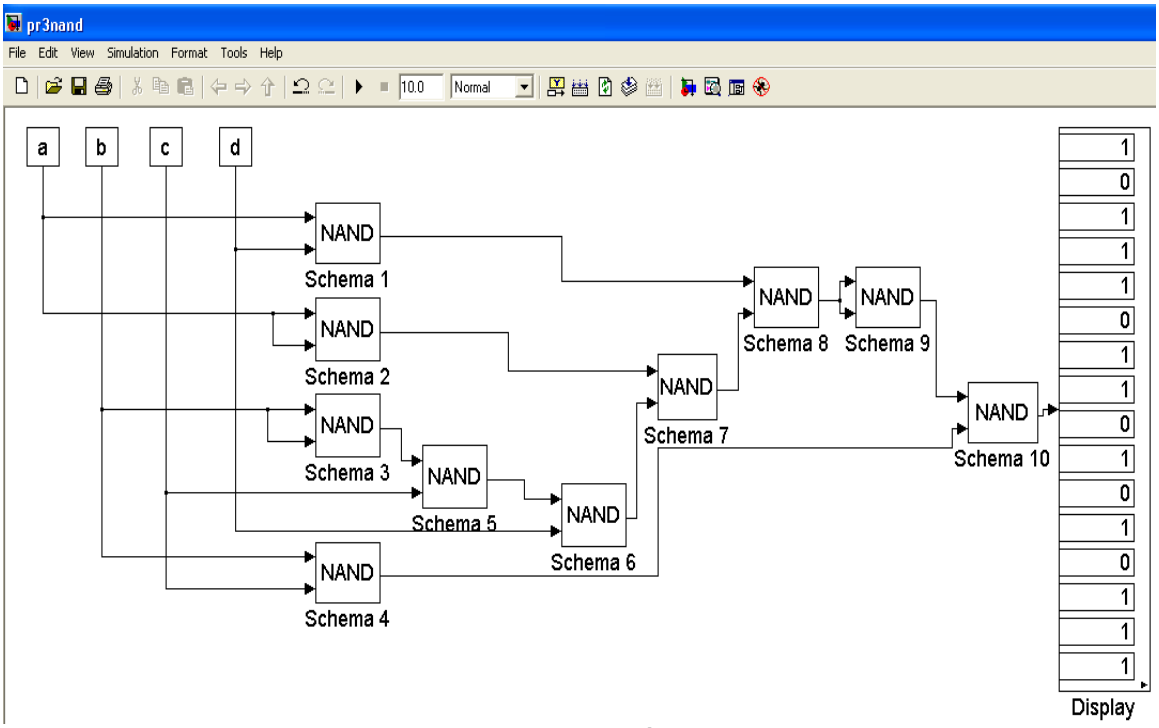
A její úprava pro realizaci kombinačním logickým obvodem prvky NAND je dána takto

$$y = \overline{\overline{ad} \cdot \overline{\bar{a} \cdot \overline{\overline{\bar{b}c} \cdot \bar{d}} \cdot \overline{bc}}} \qquad (13)$$

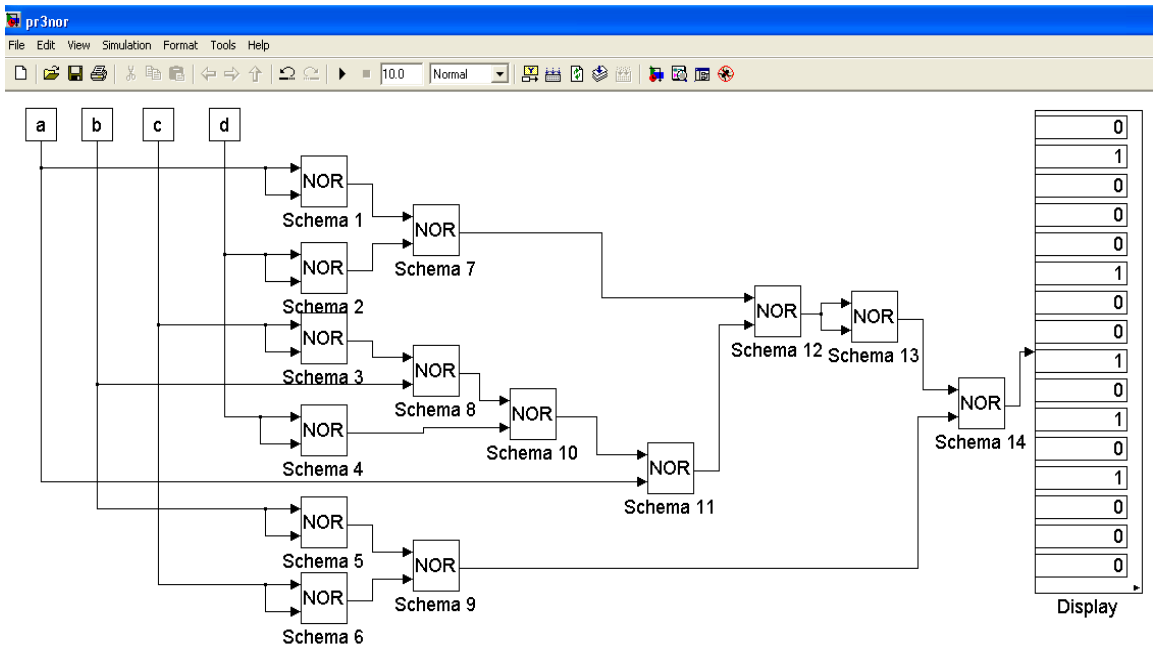
A úprava pro realizaci KLO prvky NOR je obdobně upravena jako

$$y = \overline{\overline{\bar{a} + \bar{d}}} + \overline{a + \overline{\overline{b + \bar{c}} + \bar{d}}} + \overline{\overline{\bar{b} + \bar{c}}} \qquad (14)$$

Přiklad 3 je realizovány v programu Matlab/Simulink z NAND prvků (obr. 7) a schéma z NOR prvků (obr. 8)



Obr. 7. Příklad 3 – schéma z NAND prvků



Obr. 8. Příklad 3 – schéma z NOR prvků

Pravdivostní tabulka pro logickou funkci (12) musí odpovídat hodnotám zobrazeným v bloku Display, viz (obr. 7) a (obr. 8).

Tab. 7 Pravdivostní tabulka pro logickou funkci viz (9)

a	b	c	d	y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

4. Tvorba aplikace v programu Matlab/Guide

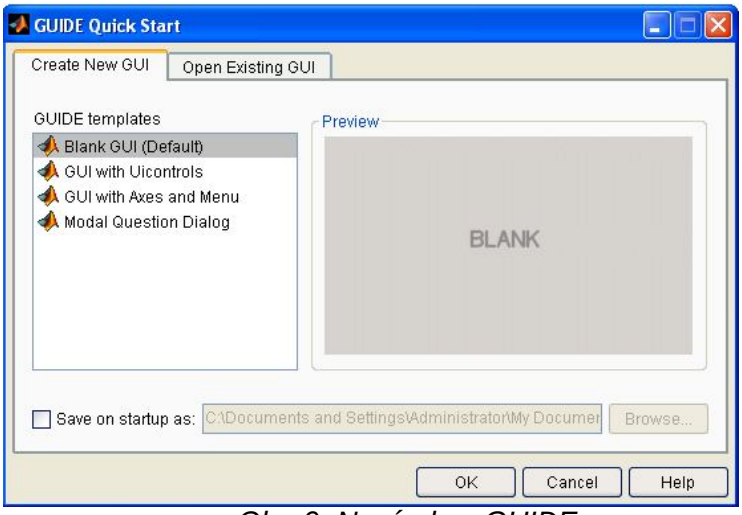
Tvorba GUI je součástí vývojových a programovacích nástrojů, které MATLAB poskytuje.

4.1 Popis grafického uživatelského prostředí MATLAB/GUIDE

GUIDE je MATLAB vývojové prostředí pro tvorbu GUI (grafický uživatelský interface) také umožňuje vytvářet a editovat uživatelský interface a to prostřednictvím list box, pull-down menu, push button, radio buton, checkbox, sliders a dalších. Funkce GUIDE spouští vývojové prostředí pro vytváření GUI. Zde můžeme vybrat a rozmístit požadované ovládací a grafické objekty (osy, čáry, plochy, . . .). Každý objekt má své jedinečné číslo (handle) a své vlastnosti. Lze tedy nastavovat a měnit jejich vlastnosti. GUIDE vytváří soubor s příponou fig, kde jsou zapsány informace o grafickém okně a soubor s příponou m, který lze editovat a programovat tak „CALLBACK“ jednotlivých funkcí.
Funkční GUI se generuje aktivováním návrhu vytvořeného v Layout Editoru vybráním. Activate Figure položky v Tools menu nebo klikáním na ikonu v toolbar. Odehraje se procedura, při které GUIDE uloží M-file a FIG-file. Jestliže jste ještě návrh neuložili, otevře se Save As dialog a zeptá se na název generovaného M-file. Po zadání jej uloží společně s příslušným FIG-file. MATLAB pak vykoná generovaný M-file k zobrazení GUI.

Guide Existuje mnoho způsobů, jak spustit Guide. Můžete začít GUIDE od:

- Příkazový řádek zadáním příkazu
- Start menu Guide výběrem MATLAB> Guide (GUI Builder)
- MATLAB menu File výběrem New> GUI
- MATLAB nástrojové liště kliknutím na tlačítko GUIDE



Obr. 9. Nové okno GUIDE

V GUIDE Quick Start dialogové okno obsahuje dvě záložky:

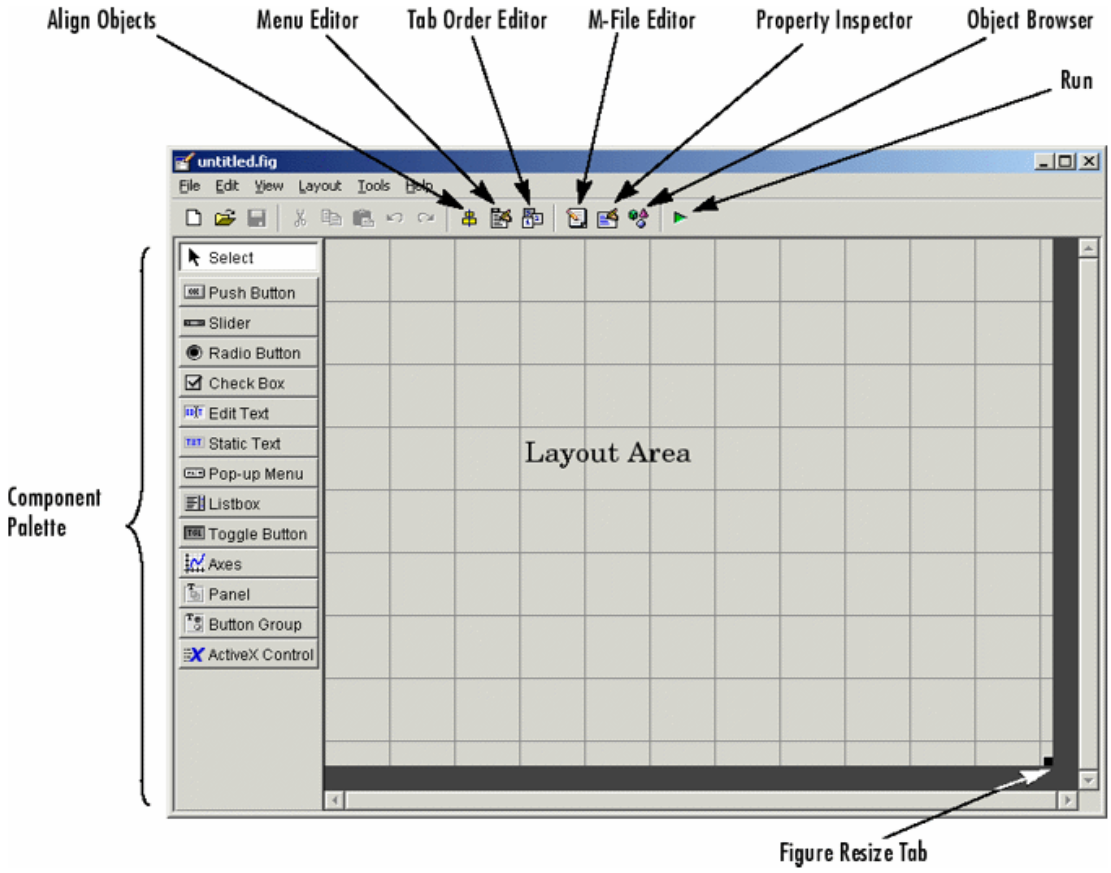
- Vytvořit nový GUI - vytvoření nového GUI a to výběrem ze šablon nebo už dávným uložený GUI a to tak, že dáme otevřít stávající GUI.
- Chcete-li otevřít existující GUI z příručky tak vybíráme z aktuálního adresáře nebo z jiné.

[Matlab GUI]

4.2 Funkci v souvislosti s problematikou generovaní bloků Simulink

Shrnutí nástroje GUIDE:

V GUIDE jsou k dispozici nástroje od Layout Editor které jsou znázorněny na obr. 10. Nástroje jsou uvedeny v tabulce a stručně popsány a ukázány, jak se s nimi pracuje.



Obr. 10. Seznam ovládacích prvků Paletě nástrojů prostředí GUIDE

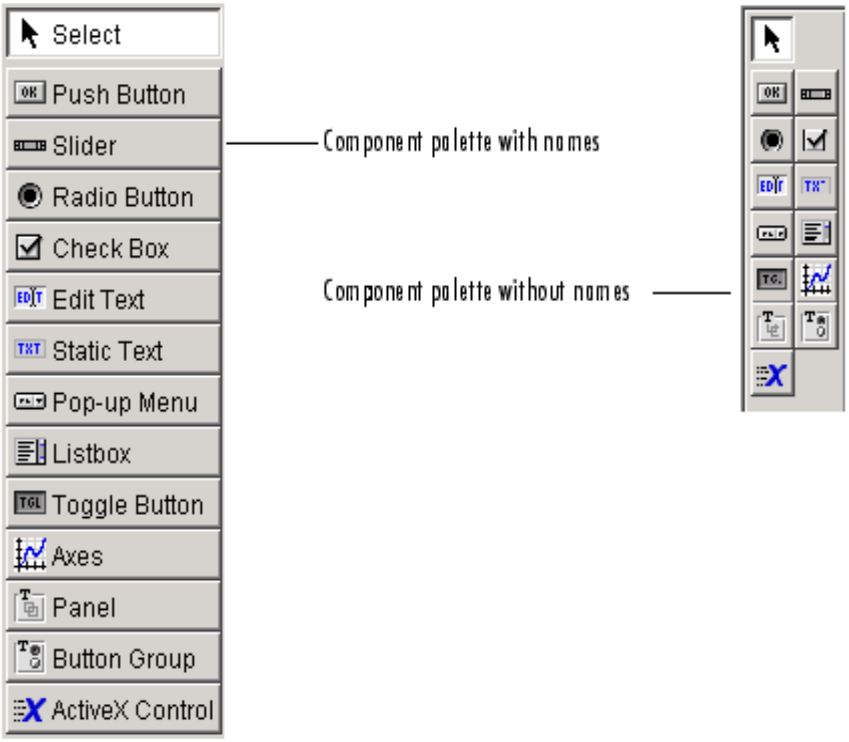
Tab. 8 Popis nabídky aplikace GUI

Nabídky aplikace Gui	**Popis**
Návrhové nástroje	Vysvětlení nástroje
Layout Editor	Ovládací panel pro GUIDE, k jeho startu slouží příkaz guide, umožňuje vybírat GUI komponenty z palety a uspořádat je v okně pro návrh.
Figure Resize Tab	Přizpůsobuje rozměry pracovní oblasti.
Menu Editor	Vytváří řádkové menu okna a kontextové menu.
Align Objects	Nástroj pro zarovnání grafických objektů (s ohledem na ostatní objekty.

Nabídky aplikace Gui	**Popis**
Tab Order Editor	Mění pořadí, ve kterém jsou prvky vybrané tabulátorem.
Property Inspector	Umožňuje nastavovat vlastnosti komponent v návrhu a zobrazuje jejich aktuální hodnoty.
Object Browser	Zobrazuje hierarchickou strukturu objektů v návrhu.
Run	Spuštění celkové vytvořené aplikace.
M-File Editor	Ukládání souborů na disk.

Dostupné komponenty

Složka paletě na levé straně v Layout Editor obsahuje složky, které můžete přidat do svého GUI.



Obr. 11. Seznam tlačítky v paletě nástrojů v prostředí Guide

Lišta s ovládacími prvky (component palette-*komponent paleta*) obsahuje všechny možné objekty, které je možné do dialogu vložit.

Tab. 9 Popis ovládací prvků

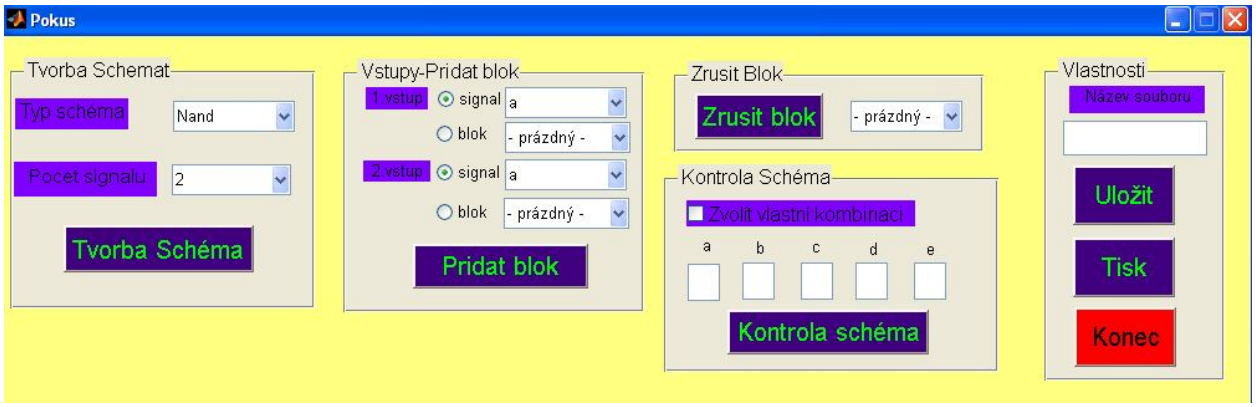
Komponent	Ikona	Popis
Push Button¨ (Tisknutelné tlačítko)		Je v grafickém uživatelském rozhraní pravděpodobně jeden z nejpoužívanějších ovládacích prvků, který dovoluje uživateli v aplikaci spustit nějakou určitou událost.
Toggle Button (Přepínací tlačítko)		Označuje stav, například Ano/Ne, nebo režim, například Zapnuto/Vypnuto. Kliknutím na toto tlačítko přepínáte mezi stavem zapnuto a vypnuto. Pomocí přepínacího tlačítka lze například přepínat mezi režimem návrhu a režimem úprav nebo je lze použít jako alternativu k zaškrtávacímu políčku. Přepínací tlačítko není k dispozici jako ovládací prvek formuláře, pouze jako ovládací prvek ActiveX.
Radio Button (Přepínač)		Ovládací prvek umožňující uživatelům vybírat z několika možností, které se vzájemně vylučují. Je-li vybrán jeden přepínač ve skupině, ostatní přepínače nejsou vybrány. Skupina přepínačů je vázána na jedno pole ve zdroji dat a každý přepínač ukládá do pole jinou hodnotu.
Check Box (Zaškrtávací políčko)		Ovládací prvek umožňující nastavit hodnoty typu Ano/Ne nebo True/False pomocí zaškrtnutí nebo zrušení zaškrtnutí čtvercového políčka.
Edit Text (Editační pole)		Používá editační pole k zobrazení a modifikaci dat ze zadané položky aktuálního záznamu.
Static Text (Textové pole)		Nejčastěji používané ovládací prvky na formuláři. Do textových polí mohou uživatelé vkládat neformátovaný text, například věty, jména, čísla, kalendářní data a časové údaje. Textová pole nemohou obsahovat formátovaný text.
Slider (Šoupátko)		Klepnutím na posuvník nebo přetažením jezdce dojde k posunutí oblasti. Klepnutím mezi jezdce a posuvník je možné posunout se o jednu stránku.
List Box (Seznam)		Ovládací prvek, který obsahuje seznam položek v poli, z něhož mohou uživatelé vybrat požadovanou položku. Zobrazené položky mohou pocházet ze

Komponent	Ikona	Popis
		seznamu, který vytvoříte ručně, z hodnot ve zdroji dat formuláře nebo z hodnot získaných prostřednictvím datového připojení k dokumentu XML, databáze, webové služby nebo knihovny či seznamu služby Windows SharePoint Services.
Pop-Up Menu (Pole se seznamem)		Ovládací prvek, který obsahuje seznam položek v poli, z něhož mohou uživatelé vybrat požadovanou položku nebo zadat položku vlastní. Zobrazené položky mohou pocházet ze seznamu, který vytvoříte ručně, z hodnot ve zdroji dat formuláře nebo z hodnot získaných prostřednictvím datového připojení k dokumentu XML, databáze, webové služby nebo knihovny či seznamu služby Windows SharePoint Services.
Axes (Graf)		Vytvoří se okno, kde mohou být zobrazeny obrázky nebo grafické průběhy. Stejně jako všechny grafické objekty, os, mají vlastnosti, které můžete nastavit.
Panel		Jedná se o panel (rám), do kterého můžeme vkládat jednotlivé ovládací prvky.
Button Group (Skupina tlačítek))		Speciální panel určen pro zatrhávací tlačítka (Radio button), v tomto panelu může být zatrhnuto pouze jedno zmíněné tlačítko.
ActiveX Component (Komponenta Active X)		Pomocí tohoto prvku je možné vkládat do objektu ActiveX komponenty, umožní tak zobrazit ovládací prvky ActiveX v grafického rozhraní. Jsou k dispozici pouze na Microsoft Windows platforma.

5. Tvorba uživatelské aplikace pro tvorbu schémat NAND a NOR

Hlavním cílem bylo usnadnit studentům tvorbu schémat v prostředí MATLAB a umožnit jim schémata zjednodušit k sestavování a ověřování jejich funkčnosti na základě výsledků logické funkce po sestavení schématu.

5.1 Návrh vzhledu uživatelské aplikace



Obr. 12. Uživatelské aplikace

A. Typ schéma

Typ schématu nabízí dva typy, které si můžeme vybrat konkrétně z NAND nebo NOR

B. Počet signálů

Zde si můžeme zadat jakýkoliv počet signálů.

***C.* Tvorba schéma**

Když máme vybraný typ schématu a zadaný počet signálů klikneme na „Tvorba schéma“ a tím se vytvoří dané okno, mezitím se zablokují naše požadavky na schéma a signál. Když uživatel potřebuje vytvořit znovu schéma pak stiskne to tlačítko ještě jednou.

D. Vstupy - Přidat blok

Zde si vybíráme ze dvou vstupů, vstup 1 a vstup 2. V každém vstupu si dále můžeme zvolit signál nebo blok. Lze tedy vybrat buď ze signálů nebo z bloků. Když budeme chtít vybrat položku „blok“ v okénku se vybírá blok 1, blok 2, … to znamená, že jdou vytvořit další bloky podle signálů.

Po kliknutí na tlačítko „Přidat blok“ se vytvoří nový blok se vstupy, které si uživatel vybral.

E. Zrušit blok

Někdy student potřebuje zrušit blok, který si myslí, že je špatný a může použít tlačítko „Zrušit blok“

F. Kontrola schéma

Student má hodnoty pravdivostní tabulky pro danou funkci, kterou realizuje bezkontaktním schématem NAND nebo NOR a potřebuje zjistit, zde navržené schéma odpovídá pravdivostní tabulce.

G. Tisk

„Tisk“ umožňuje vytisknutí vytvořeného schématu pro tvorbu dokumentace.

H. Uložit

Kliknutím na „Uložit“ uložíme schémata pod zadaným názvem a příponou *.mdl.

I. Konec

Tlačítko „Konec“ slouží pro zavření GUI aplikace.

5.2 Ovládací prvek i kódů pro uživatelskou funkci

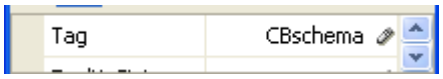
Uživatelská aplikace-ovládací prvky a zdrojový kód

A. Typ schéma

Pro výběr vstupu prvku NAND či NOR, kteří byly vytvořeny z kódu programu v m-souboru

```
typ=get(handles.CBschema,'Value')
if typ==1
add_block('built-in/Logical Operator',B,'Operator','NAND')
else
add_block('built-in/Logical Operator',B,'Operator','NOR')
end
```

U vlastnosti „Tag“ lze zvolit „Typ schéma“ s názvem „Cbschéma“



Pomocí příkazu „get“, funkce zkontroluje hodnotu „Cbschéma“.

Když je hodnota rovna 1, tak blok je vytvořený jako NAND, jinak bude blok vytvořen jako typ NOR. Příkaz „add_block“ vloží tento blok do Simulink okna.

B. Počet signál

Ovládací prvek „Pole se seznamem“ s názvem „Počet signálů“ je určen pro volbu počtu vstupů logické funkce. Tvorba tohoto ovládacího prvku je obdobná jako ovládací prvek

„Typ schéma“. Aplikace umožňuje realizovat bezkontaktní schéma pro 2, 3, 4 nebo 5 logických vstupů. Podle výběru hodnoty počtu vstupů je otevřena příslušná aplikace a toto je provedeno příkazem „switch“ tak, že uživatel stiskne tlačítko „Tvorba schéma“.

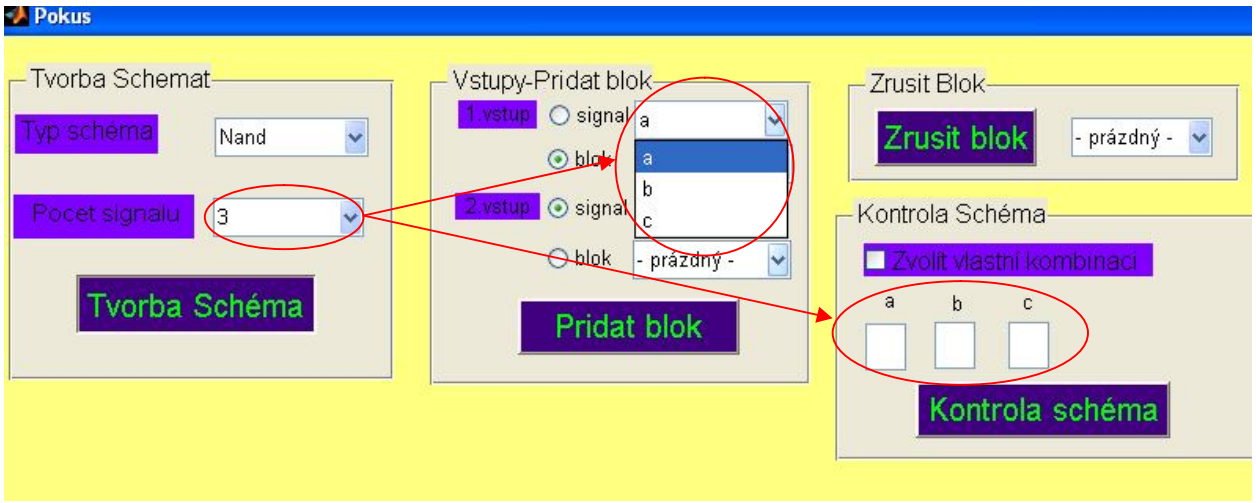
C. Tvorba schéma

Tlačítko „Tvorba schéma“ bylo realizováno ovládacím prvkem s názvem „TS“ ale není obdobná jako ovládací tlačítko „Typ schéma“ a „Počet signálu“ „Tvorba schéma“ bylo realizováno jako přepínací tlačítko. Když na to tlačítko klikem automaticky se objeví nové okno, které bylo požadavky vytvořeno a pak uživatel nemá možnost změnit hodnoty v ovládacích prvcích. A uživatel stiskne na tlačítko zpět pak se program vrátí do původního stavů a uživatel může vytvořit nové schéma. Tlačítko „Tvorba schéma“ může vytvořit Simulink okno pomocí těchto dvě příkazů „open_system“ „switch“.

```
P=get(handles.TS,'Value');
if P==1
 X=get(handles.PCvstupy,'Value');
 switch X+1
     case 2
         handles.L='start2';
         open_system ('start2.mdl');
         set(handles.S3,'Visible','off');
         set(handles.E3,'Visible','off');
         set(handles.S4,'Visible','off');
         set(handles.E4,'Visible','off');
         set(handles.S5,'Visible','off');
         set(handles.E5,'Visible','off');
         set(handles.CBsignal,'String',{'a';'b'});
         set(handles.CBsignal1,'String',{'a';'b'});
     case 3
         handles.L='start3';
         open_system ('start3.mdl');
         set(handles.S4,'Visible','off');
         set(handles.E4,'Visible','off');
         set(handles.S5,'Visible','off');
         set(handles.E5,'Visible','off');
         set(handles.CBsignal,'String',{'a';'b';'c'});
         set(handles.CBsignal1,'String',{'a';'b';'c'});
     case 4
         handles.L='start4';
         open_system ('start4.mdl');
         set(handles.S5,'Visible','off');
         set(handles.E5,'Visible','off');
         set(handles.CBsignal,'String',{'a';'b';'c';'d'});
         set(handles.CBsignal1,'String',{'a';'b';'c';'d'});
     case 5
         handles.L='start5';
```

```
            open_system ('start5.mdl');
    end
```

V tomto programu, když si uživatel vybere například hodnotu 3 v „Počet signálu“ objeví se pak v „1.vstup“ i „2.vstup“ 3 signály pojmenované a, b, c a zároveň i ve funkci „Kontrola schéma“.



Obr. 13. Program může být změněn z vhodné se volby uživatele

Po otevření okna Simulink pro vytvoření nového bezkontaktního schématu uživatel nemá možnost změnit hodnoty v ovládacích prvcích „Typ schéma“ a „Počet signálu“. To je způsobeno nastavením vlastnosti „Enable“ na hodnotu „inactive“.

```
set(handles.CBschema,'Enable','inactive')
set(handles.PCvstupy,'Enable','inactive')
```

Když uživatel už nechce pokračovat a chce vytvořit znovu pak stiskne tlačítko „Tvorba schéma“ ještě jednou a se objeví okno s názvem „Znovu?“ ve kterém je otázka „Chcete vytvořit znovu?“ uživateli se nabídne „Ano“ nebo „Ne“ výběrem „Ano“ se uživateli zobrazí další okno „Chyba“ s větou „Musíte zavřít staré Simulink okno“ a v tomto okno klikneme na „Zavřít“ tím se vrátí do původního stavů pro tvorbu nového okna (obr.14)

Obr. 14. Okna display když uživatel stiskne ještě jednou tlačítko „Tvorba schéma“

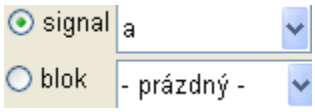
```
    Novy=questdlg('Chcete vytvorít znovu?', ...
                  'Znovu?', ...
                  'Ano','Ne','Ano');
                if strcmp(Novy,'Ne')
                return
                else
       Znovy=questdlg('Musite zavrít staré Simulink okno', ...
                  'Chyba', ...
                  'Zavrít','Zavrít');
                if strcmp(Znovy,'Zavrít')
                        set(handles.CBschema,'Enable','on');
                        set(handles.PCvstupy,'Enable','on');
                        set(handles.S3,'Visible','on');
                        set(handles.E3,'Visible','on');
                        set(handles.S4,'Visible','on');
                        set(handles.E4,'Visible','on');
                        set(handles.S5,'Visible','on');
                        set(handles.E5,'Visible','on');

set(handles.CBsignal,'String',{'a';'b';'c';'d';'e'}');

set(handles.CBsignal1,'String',{'a';'b';'c';'d';'e'}');
                set(handles.CBzrusit,'Value',1);
                set(handles.CBzrusit,'String','- prázdný -');
                set(handles.CBblok,'Value',1);
                set(handles.CBblok,'String','- prázdný -');
                set(handles.CBblok1,'Value',1);
                set(handles.CBblok1,'String','- prázdný -');
```

```
                end
                end
    end
```

Když uživatel zavře předešlé dvě okna pak už má možnost změnit hodnoty v ovládacích prvcích „Typ schéma“ a „Počet signálů“ to je způsobeno nastavením vlastnosti „Enable“ na hodnotu „on“.

D. Vstupy - Přidat blok

V tomto okně jsou dva vstupy ty se dále děli na blok a signál což jsou „Radio button“ ale lze označit jen signál, jelikož blok je „- prázdný -“ a tím nejde označit.

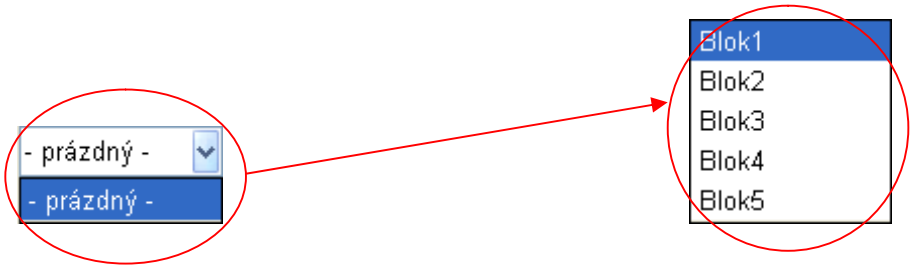


Kódy jsou:

```
    function varargout = blok_Callback(h, eventdata, handles,
varargin)
    signal=get(handles.signal,'Value');
    blok=get(handles.blok,'Value');
    if blok==1
       signal=0;
       set(handles.blok, 'Value', 1), on = (handles.blok);
       set(handles.signal,'Value', 0), off = (handles.signal);
       if get(handles.CBblok,'String')=='- prázdný -'
        blok==0
        signal=1;
      set(handles.blok, 'Value', 0), off = (handles.blok);
       set(handles.signal,'Value', 1),  on = (handles.signal)
       end
    elseif blok==0
        signal=1;
      set(handles.blok, 'Value', 0), off = (handles.blok);
       set(handles.signal,'Value', 1),  on = (handles.signal);
    end
    function varargout =signal_Callback(h, eventdata, handles,
varargin)
    blok=get(handles.blok,'Value');
    signal=get(handles.signal,'Value');
    if signal==1
       blok=0;
       set(handles.signal, 'Value', 1), on = (handles.signal);
       set(handles.blok,'Value', 0), off = (handles.blok);
    elseif signal==0
        blok=1;
       set(handles.signal, 'Value', 0), off = (handles.signal);
```

```
        set(handles.blok,'Value', 1),   on = (handles.blok);
    end
```

Kliknutím na „Přidat blok“ se uvolní přepínací tlačítko blok a v jeho „Pole se seznamem“ se objeví tolik bloků kolik krát stiskni tlačítko „Přidat blok“



Kódy:

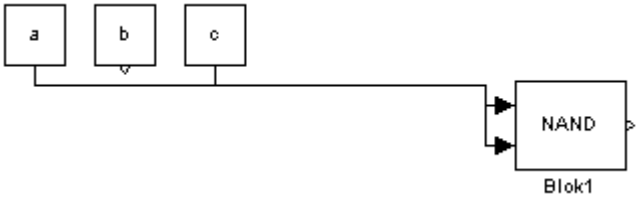
```
    text='Blok';
    S=get(handles.CBblok,'String');
    if strcmp(S,'- prázdný -')
       set(handles.CBblok,'String','Blok1');
       B=[handles.L '/Blok1'];
       nazev1=['Blok1' '/1'];
       nazev2=['Blok1' '/2'];
    else
        posledni=size(S,1);
        if posledni==1
            text=S;
            set(handles.CBblok,'String',{'Blok1';'Blok2'});
            B=[handles.L '/Blok2'];
            nazev1=['Blok2' '/1'];
            nazev2=['Blok2' '/2'];
        else
            text=S{posledni,1};
            if size(text,2)==5
                poradi=text(1,5);

            else
                poradi=text(1,5:6);
            end
            poradiCislo=str2num(poradi);
            poradiCislo=poradiCislo+1;
            S{posledni+1,1}=['Blok' num2str(poradiCislo)];
            set(handles.CBblok,'String',S);
            B=[handles.L '/Blok' num2str(poradiCislo)];
            nazev1=['Blok' num2str(poradiCislo) '/1'];
            nazev2=['Blok' num2str(poradiCislo) '/2'];
        end
    end
```

Výběr 2.vstupu je proveden obdobným způsobem jako volba 1.vstupu

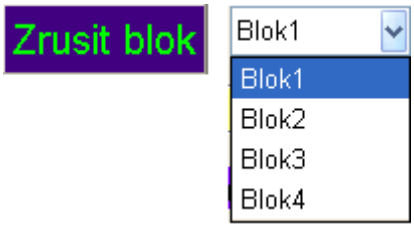
Funkce pro tlačítko „Přidat blok“ zjistí aktuální nastavení obou vstupů pomocí funkce „get“ a podle nich generuje blok „Logical Operator“, dále přidá spojovací čáru do bloku, a přidá pozici bloků.

```
vstup1B=get(handles.blok,'Value');
  if vstup1B==1
     Blok=get(handles.CBblok,'String');
     cisloB=get(handles.CBblok,'Value');
     R=Blok(cisloB,1);
     SB1=[R{1,1} '/1'];
     % ulozit do handles
     handles.vstup1{str2num(nazev1(1,5)),1}=R{1,1};
 else
     Vstup=get(handles.CBsignal,'String');
     AktualRadek=get(handles.CBsignal,'Value');
     vstupB=Vstup(AktualRadek,1);
     SB1=[vstupB{1,1} '/1'];
     % ulozit do handles
     handles.vstup1{str2num(nazev1(1,5)),1}=vstupB{1,1};
 end
 vstup2B=get(handles.blok1,'Value');
 if vstup2B==1
     Blok1=get(handles.CBblok1,'String');
     cisloB1=get(handles.CBblok1,'Value');
     R1=Blok1(cisloB1,1);
     SB2=[R1{1,1} '/1'];
      % ulozit do handles
     handles.vstup2{str2num(nazev2(1,5)),1}=R1{1,1};
 else
     Vstup=get(handles.CBsignal1,'String');
     AktualRadek=get(handles.CBsignal1,'Value');
     vstupB=Vstup(AktualRadek,1);
     SB2=[vstupB{1,1} '/1'];
     % ulozit do handles
     handles.vstup2{str2num(nazev2(1,5)),1}=vstupB{1,1};
 end
 add_line(handles.L,SB1,nazev1,'autorouting','on')
 add_line(handles.L,SB2,nazev2,'autorouting','on')
 guidata(hObject, handles);
```

Jestliže uživatel chce přidat blok NAND ze signálu a i c tak v okně Simulink se zobrazí toto schéma

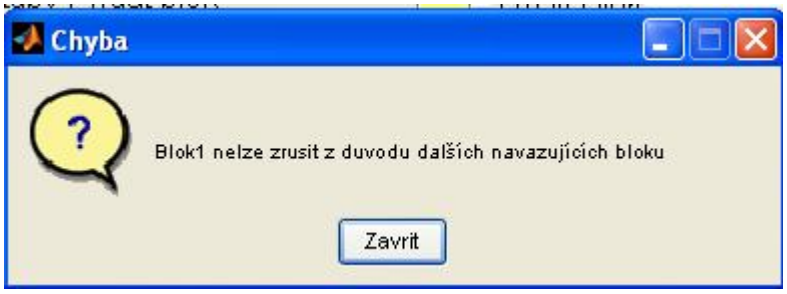


F. Zrušit blok



Funkce: „Zrušit blok“ obsahuje 2 ovládací prvky. 1 je tlačítko „Zrušit blok“ 2 je „Pole se seznamem“ kde display bloků v schéma na pomoc uživateli muže vybrat bloků na zrušit.

Uživatel nemůže zrušit bloků který z důvodu dalších navazujících bloku (obr. 15)



Obr. 15. Okno chyba z důvodu blok nelze zrušit

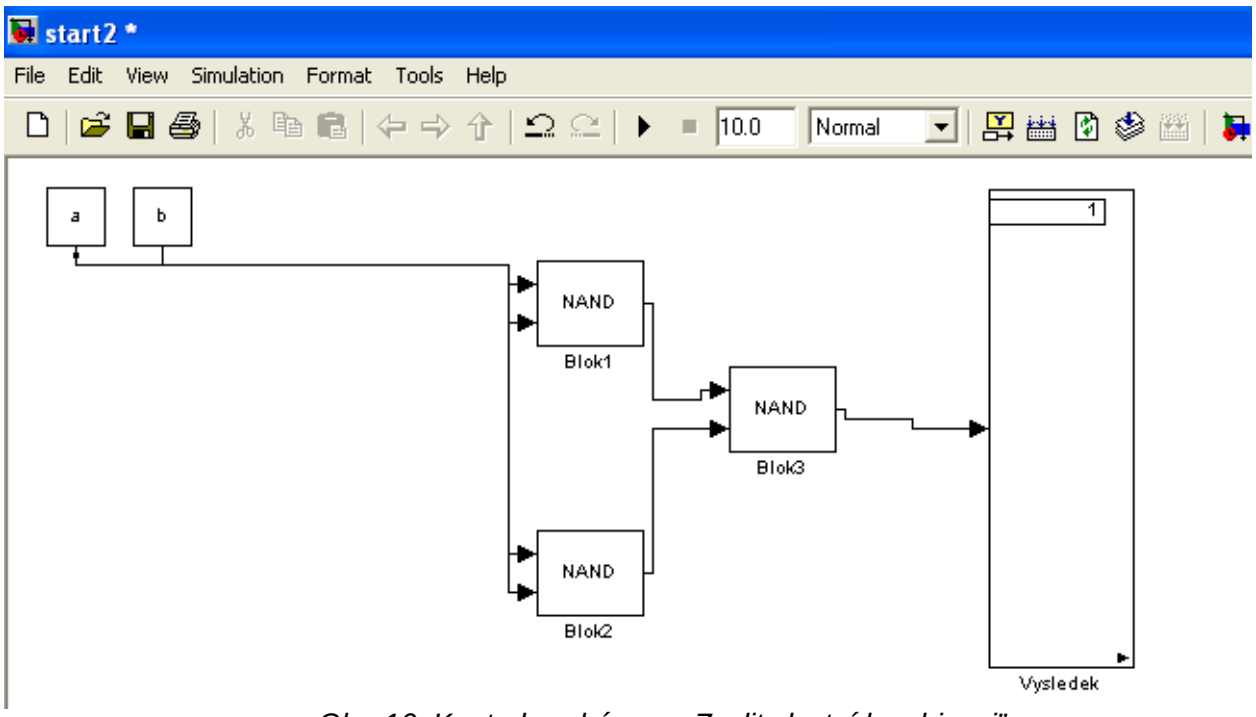
Zde jsou kódy pro zrušení bloků:

```
function PBzrusit_Callback(hObject, eventdata, handles)
X=get(handles.CBzrusit,'String');
Y=get(handles.CBzrusit,'Value');
Z=X(Y,1);
if Y==1
    pom='Blok1'
else
   pom=Z{1,1};
end
cisloRad=str2num(pom(1,5));
vstup1=handles.vstup1{cisloRad,1};
vstup2=handles.vstup2{cisloRad,1};
n=size(handles.vstup1,1)
for i=1:n
    if handles.vstup1{i,1}==pom;
       X=questdlg('Blok1 nelze zrusit z duvodu dalších
navazujících bloku', ...
                  'Chyba', ...
```

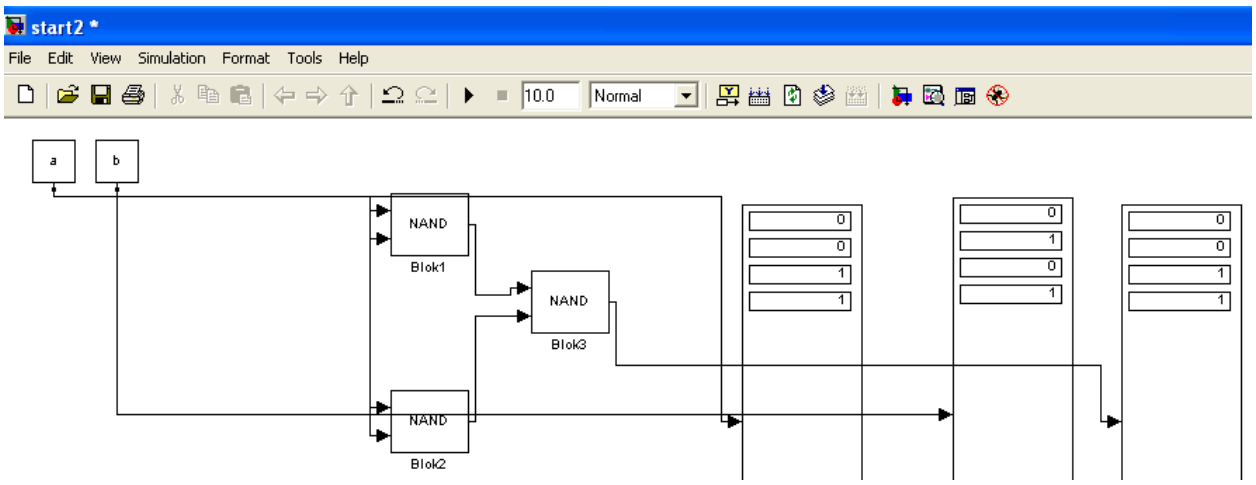
```
                    'Zavrit','Zavrit');
                if strcmp(X,'Zavrit')
                return
                end
        end
        if handles.vstup2{i,1}==pom;
           X=questdlg('Blok1 nelze zrusit z duvodu dalších
navazujících bloku', ...
                    'Chyba', ...
                    'Zavrit','Zavrit','Zavrit');
                if strcmp(X,'Zavrit')
                return
                end
        end
    end
    delete_line(handles.L, [vstup1 '/1'],[pom '/1']);
    delete_line(handles.L, [vstup2 '/1'],[pom '/2']);
    delete_block([handles.L '/' pom]);
    if cisloRad==1
            A1='- prázdný -';
    else
        for i=1:cisloRad-1,
             A1{i,1}=X{i,1};
        end
        konec=size(X,1);
        if konec==cisloRad
        else
            for i=cisloRad+1:konec,
                A1{i-1,1}=X{i,1};
            end
        end
    end
    % nahradit retezec v popup menu
    set(handles.CBzrusit,'Value',1);
    set(handles.CBzrusit,'String',A1);
    set(handles.CBblok,'Value',1);
    set(handles.CBblok,'String',A1);
    set(handles.CBblok1,'Value',1);
    set(handles.CBblok1,'String',A1);
```

G. Kontrola schéma

Uživatel má 2 způsoby na „Kontrola schéma“. První způsob je kontrola s „Zvolit vlastní kombinaci“, to znamená jestliže uživatel zaškrtne pole pak dá signálu jednu hodnotu (0 nebo 1) a program bude kontrolovat podle té hodnoty (obr.16). Jinak když uživatel nezaškrtne pole „Zvolit vlastní kombinaci“ pak program zkontroluje celé schéma. (obr.17).



Obr. 16. Kontrola schéma s „Zvolit vlastní kombinaci"



Obr. 17. Kontrola schéma bez „Zvolit vlastní kombinaci"

Kódy pro funkci „Kontrola schéma"

```
function PBKontrola_Callback(hObject, eventdata, handles)
O=get(handles.CheckBox,'Value');
if O==1
    if handles.L=='start4';
    A=get(handles.E1,'String');
    B=get(handles.E2,'String');
    C=get(handles.E3,'String');
```

```
         D=get(handles.E4,'String');
         aS=boolean([str2num(A)])
         assignin('base','a',aS)
         bS=boolean([str2num(B)])
         assignin('base','b',bS)
         cS=boolean([str2num(C)])
         assignin('base','c',cS)
         dS=boolean([str2num(D)])
         assignin('base','d',dS)
         end
        if handles.L=='start4';
                aS=boolean([0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]);
                assignin('base','a',aS);
                bS=boolean([0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1]);
                assignin('base','b',bS);
                cS=boolean([0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1]);
                assignin('base','c',cS);
                dS=boolean([0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1]);
                assignin('base','d',dS);
        add_block('built-in/Display',[handles.L '/Vstup a'])
        add_block('built-in/Display',[handles.L '/Vstup b'])
        add_block('built-in/Display',[handles.L '/Vstup c'])
        add_block('built-in/Display',[handles.L '/Vstup d'])
        add_line(handles.L,'a/1','Vstup a/1','autorouting','on')
        add_line(handles.L,'b/1','Vstup b/1','autorouting','on')
        add_line(handles.L,'c/1','Vstup c/1','autorouting','on')
        add_line(handles.L,'d/1','Vstup d/1','autorouting','on')
        set_param([handles.L '/Vstup a'],'Position',[675, 61,
760, 369])
        set_param([handles.L '/Vstup b'],'Position',[775, 56,
860, 364])
        set_param([handles.L '/Vstup c'],'Position',[875, 56,
960, 364])
        set_param([handles.L '/Vstup d'],'Position',[975, 56,
1060, 364])
        end
        add_block('built-in/Display',[handles.L '/Vysledek'])
        set_param([handles.L '/Vysledek'],'Position',[1175, 56,
1260, 364])
        add_line(handles.L, [handles.nazev
'/1'],'Vysledek/1','autorouting','on')
        sim (handles.L)
```

H. Tisk

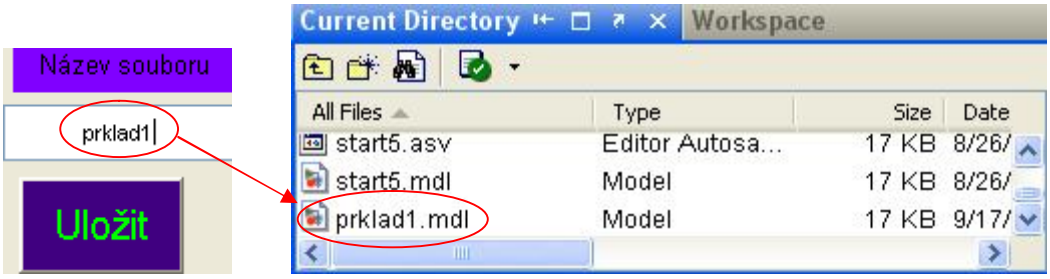
Pro volbu „tisk“ byl vytvořen jednoduché kód zde je:

```
Print -s
```

Uživatel také může použit jednoduše, kliknout na to tlačítko pak se schéma vytiskne na papír.

I. Uložit

Uživatel může použit funkci „Uložit“ jednoduše, ta funkce má „Editační pole“ pro uživatele aby nazvali soubor, pak stiskne „Uložit“ a má schéma uložené. Například uživatel nazve „přiklad 1“ pak bude mít uloženo jako (obr 18)

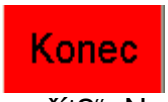


Obr. 18. „priklad1“ uloženo z funkce „Uložit“

Zde jsou kódy:

```
function PBUlozit_Callback(hObject, eventdata, handles)
S=get(handles.edit9,'String');
   SAVE_SYSTEM(handles.L,S)
```

J. Konec



Poslední funkce je „Konec“. Když uživatel chce ukončit program, stiskne na tlačítko „Konec“ zobrazí se na display jedno okno a zeptá se: „Chcete zavřít?“. Na výběr má „Ano“ nebo „Ne“ jestliže uživatel stiskne „Ano“ pak program bude zavřen a když „Ne“ pak se vrátí do programu (obr.19).

Obr. 19. „Okno „Konec?“

Kódy pro funkci „Konec“

```
function Konec_Callback(hObject, eventdata, handles)
Konec=questdlg('Chcete zavrít?', ...
                'Konec?', ...
                'Ano','Ne','Ano');
```

```
if strcmp(Konec,'Ne')
    return;
else
    Close;
end
```

6. Závěr

Na začátku bakalářské práce je popsána kombinační logika dále co jsou logické funkce, logické zákony, Karnauhova mapa a kombinační logické obvody pomocí zadaných příkladů a tvorbou bezkontaktních schémat (NAND, NOR).

Další kapitola je věnována seznámení s prostřední Matlab/Simulink, kdy pro zadané příklady byly navrženy a realizovány bezkontaktní schémata. Pro každý příklad logické funkce po úpravě minimalizovaného tvaru logické funkce bylo realizováno NAND nebo NOR schéma. Tímto jsem zjistil, jaké bloky budou potřebné pro automatické generování bloků z grafické uživatelské aplikace a seznámil jsem se z jejich parametry

Podstatnou část bakalářské práce je záměrná na tvorbu aplikace v programu Matlab/GUI i v grafickém uživatelském rozhraní. Aplikace je určena studentům, kteří se neseznámili s programem Simulink a kteří potřebují realizovat bezkontaktních schémata pro kontrolu jejich správné funkčnosti vzhledem k zadané logické funkci. Aplikace je určena pro realizaci bezkontaktních schémat pro dva až pět vstupů logické funkce, což je pro daný účel postačující. Ovládací prvky grafické uživatelské aplikace umocní její jednoduché a intuitivní ovládací. Cílem aplikace je tvorba bezkontaktních schémat, kontrola realizované logické funkci pomocí všech kombinací vstupů a výsledné hodnoty výstupu logické funkce pomocí okna Display. Student pak své schéma může uložit a tisknout pro své další účely.

V práci je popsán postup tvorby grafické uživatelské aplikace a její ovládání. Automatické generování bloků z GUI prostředí program Matlab je nestandardní činností, která byla využila k propojení grafické uživatelské aplikace a výsledného blokového schématu v okně programu Simulink.

7. Literatura

Chlebek, D.2007 Vzdělávací kurz pro Základy automatizace s využitím prostředků e-learningu. Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 56 stran. Bakalářská práce, vedoucí: Wagnerová, R.

Doňar, B.; Zaplatílek, K. 2004 MATLAB - tvorba uživatelských aplikací. 2. Díl. Praha : BEN - Technická literatura, 216 p. ISBN 80-7300-133-0.

Dušek, F. 2000 MATLAB a SIMULINK: úvod do používání. Pardubice : Univerzita Pardubice, 172 s. ISBN 80-7194-776-8.

TŮMA, J., et al. 2007 Základy automatizace [online]. 1. Ostrava : Ediční středisko VŠB – TUO, Available from www: <http://www.elearn.vsb.cz/archivcd/FS/Zaut/Skripta_text.pdf>. 978-80-248-1523-7.

MATLAB Graphical User Interface for EE Students [online]. Available from www: <http://people.msoe.edu/~saadat/matlabgui.htm>

GUIDE, the Matlab gui maker [online]. Available from www: <http://www.uchsc.edu/physiology/ wjb/matlab/matlab2003c.htm>

Matlab GUI [online]. Available from www: < http://instruct1.cit.cornell.edu/courses/bionb441/ GUIdesign/guiprog.html>

Logické řízení [online]. Available from www: <http://195.178.89.122/CAAC_PHP//CAAC/cesky/logrizen/ztklo /ztklo.php>

Wikipedia [online]. Available from www: http://www.wikipedia.org/